

Projeto de Bases de Dados

Parte 3

Grupo: 50			
Número	Nome	Contribuição	Esforço (em Horas)
93695	Catarina Sofia dos Santos Sousa	33%	5h
93743	Nelson Alexandre Geada Trindade	34%	5h
93754	Rodrigo Rodrigues Major	33%	5h

Sala: 1-29

Turno: 4ªFeira – 15h00

Docente: Carlota De Oliveira Lopes Dias

Criação da Base de Dados

```
-- #####
-- ##### CREATE TABLES #####
-- #####
-- AUXILIAR Tables
CREATE TABLE nome_concelho ( nome varchar(24) NOT NULL UNIQUE PRIMARY KEY );
CREATE TABLE nome_regiao ( nome varchar(8) NOT NULL UNIQUE PRIMARY KEY );
CREATE TABLE tipo_instituicao ( tipo varchar(11) PRIMARY KEY );
REVOKE ALL PRIVILEGES ON Table nome_concelho FROM public;
REVOKE ALL PRIVILEGES ON Table nome_regiao FROM public;
REVOKE ALL PRIVILEGES ON Table tipo_instituicao FROM public;

-- Principal Tables
CREATE TABLE regiao (
    num_regiao int PRIMARY KEY,
    nome varchar(8) NOT NULL UNIQUE,
    num_habitantes double precision,
    UNIQUE(num_regiao, nome),
    FOREIGN KEY (nome) REFERENCES nome_regiao(nome)
);
CREATE TABLE concelho (
    num_concelho int,
    num_regiao int,
    nome varchar(24),
    num_habitantes double precision,
    PRIMARY KEY (num_concelho, num_regiao),
    FOREIGN KEY (num_regiao) REFERENCES regiao(num_regiao),
    FOREIGN KEY (nome) REFERENCES nome_concelho(nome)
);
REVOKE ALL PRIVILEGES ON Table regiao FROM public;
REVOKE ALL PRIVILEGES ON Table concelho FROM public;

CREATE TABLE instituicao (
    nome varchar(255) PRIMARY KEY,
    tipo varchar(11) NOT NULL,
    num_regiao int NOT NULL,
    num_concelho int NOT NULL,
    FOREIGN KEY (num_concelho,num_regiao)
        REFERENCES concelho(num_concelho,num_regiao),
    FOREIGN KEY (tipo) REFERENCES tipo_instituicao(tipo)
);
CREATE TABLE medico (
    num_cedula int PRIMARY KEY,
    nome varchar(255),
    especialidade varchar(25)
);
```

```

CREATE TABLE consulta (
  num_cedula int,
  num_doente int,
  data date,
  nome_instituicao varchar(255),
  CHECK (EXTRACT(ISODOW from data) < 6), --RI-consulta-1
  UNIQUE (num_doente, data, nome_instituicao), --RI-consulta-2
  PRIMARY KEY (num_cedula, num_doente, data),
  FOREIGN KEY (num_cedula) REFERENCES medico(num_cedula),
  FOREIGN KEY (nome_instituicao) REFERENCES instituicao(nome)
);

CREATE TABLE prescricao (
  num_cedula int,
  num_doente int,
  data date,
  substancia varchar(255),
  quant int,
  PRIMARY KEY (num_cedula, num_doente, data, substancia),
  FOREIGN KEY (num_cedula, num_doente, data)
    REFERENCES consulta(num_cedula, num_doente, data)
);

CREATE TABLE analise (
  num_analise int PRIMARY KEY,
  especialidade varchar(25),
  num_cedula int,
  num_doente int,
  data date,
  data_registro date,
  nome varchar(255),
  quant int,
  inst varchar(255),
  FOREIGN KEY (num_cedula, num_doente, data)
    REFERENCES consulta(num_cedula, num_doente, data),
  FOREIGN KEY (inst) REFERENCES instituicao(nome)
  --RI: a consulta associada pode estar omissa; não estando,
  -- a especialidade da consulta tem de ser igual à do médico.
);

CREATE TABLE venda_farmacia (
  num_venda int PRIMARY KEY,
  data_registro date,
  substancia varchar(255),
  quant int,
  preco int,
  inst varchar(255),
  FOREIGN KEY (inst) REFERENCES instituicao(nome)
);

```

```
CREATE TABLE prescricao_venda (
  num_cedula int,
  num_doente int,
  data date,
  substancia varchar(255),
  num_venda int,
  PRIMARY KEY (num_cedula, num_doente, data, substancia, num_venda),
  FOREIGN KEY (num_cedula, num_doente, data, substancia)
    REFERENCES prescricao(num_cedula, num_doente, data, substancia)
);
```

Consultas em SQL

Query 1

- Para os testes: Em vez de CURRENT_DATE, utilizar data = '2020-11-16';

```
-- QUERY 1
-- Qual o concelho onde se fez o maior volume de vendas hoje?
SELECT i.num_concelho
FROM instituicao i
  INNER JOIN venda_farmacia v
    ON v.inst = i.nome
WHERE v.data_registo = CURRENT_DATE
GROUP BY i.num_concelho
HAVING SUM(preco) >= all(
  SELECT SUM(preco)
  FROM instituicao i
    INNER JOIN venda_farmacia v
      ON i.nome = v.inst
  WHERE v.data_registo = CURRENT_DATE
  GROUP BY num_concelho);
```

Query 2

```
-- QUERY 2
-- Qual o médico que mais prescreveu no 1º semestre de 2019 em cada região?
With temp as (
    SELECT num_regiao, num_cedula, COUNT(num_doente) as count
    FROM prescricao
    NATURAL JOIN consulta
    INNER JOIN instituicao
    ON consulta.nome_instituicao = instituicao.nome
    WHERE prescricao.data >= '2019/01/01' and prescricao.data <= '2019/06/30'
    GROUP BY num_regiao,num_cedula
    ORDER BY num_regiao ASC
)

SELECT num_regiao, num_cedula, count
FROM temp sub
    NATURAL JOIN
    (SELECT num_regiao,MAX(sub.count) as count
     FROM temp sub GROUP BY num_regiao) sub2;
```

Query 3

```
-- QUERY 3
-- Quais são os médicos que já prescreveram aspirina em receitas aviadas em todas
-- as farmácias do concelho de Arouca este ano?
SELECT prescricao_venda.num_cedula
FROM prescricao_venda
    NATURAL JOIN venda_farmacia
    INNER JOIN instituicao
    ON venda_farmacia.inst = instituicao.nome
WHERE prescricao_venda.substancia = 'aspirina'
    AND instituicao.tipo = 'farmacia'
    AND instituicao.num_concelho = 34
    AND EXTRACT(YEAR from prescricao_venda.data) = EXTRACT(YEAR from CURRENT_DATE)
GROUP BY prescricao_venda.num_cedula
HAVING COUNT(DISTINCT instituicao.nome) >= all(
    SELECT COUNT(instituicao.nome)
    FROM instituicao
    WHERE instituicao.tipo = 'farmacia'
    AND instituicao.num_concelho = 34
);
```

Query 4

- Para os testes: utilizar `EXTRACT(MONTH from date '2019-5-1')` e `EXTRACT(YEAR from date '2019-5-1')`;

```
-- QUERY 4
-- Quais são os doentes que já fizeram análises mas ainda não aviaram prescri-
ções este mês?
SELECT DISTINCT num_doente
FROM analise
WHERE EXTRACT(MONTH from data_registro) = EXTRACT(MONTH from CURRENT_DATE)
      AND EXTRACT(YEAR from data_registro) = EXTRACT(YEAR from CURRENT_DATE)
      AND (num_cedula, num_doente, data) NOT IN
      (
        SELECT num_cedula, num_doente, data
        FROM prescricao_venda
        WHERE EXTRACT(MONTH from data) = EXTRACT(MONTH from CURRENT_DATE)
              AND EXTRACT(YEAR from data) = EXTRACT(YEAR from CURRENT_DATE)
      );
```